



AF
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Nadj et al.

Application No: 09/931,841

Filed: August 16, 2001

For: System and Method for Scheduling
and Arbitrating Events in Computing
and Networking

) Docket No: ALTEP072
)
)
) Group Art Unit: 2163
)
) Examiner: Filipczyk, Marchin R.
)
)
) Date: July 11, 2008

CERTIFICATE OF MAILING
I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA on July 11, 2008:
Signed: <i>Cynthia C. Lorente</i> Cynthia C. Lorente

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION – 37 CFR 41.37)

Commissioner for Patents
Box: Board of Patent Appeals & Interferences
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is in furtherance of the Notice of Appeal filed in this case on April 25, 2008.

This application is on behalf of:

☐ Small Entity ☒ Large Entity

Pursuant to 37 CFR 41.20(b)(2) the fee for filing the Appeal Brief is:

☐ \$255.00 (Small Entity) ☒ \$510.00 (Large Entity)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply:

☐ Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a) for the total number of months checked below:

<u>Months</u>	<u>Large Entity</u>	<u>Small Entity</u>
<input type="checkbox"/> one	\$120.00	\$60.00
<input type="checkbox"/> two	\$460.00	\$230.00
<input type="checkbox"/> three	\$1,050.00	\$525.00
<input type="checkbox"/> four	\$1,640.00	\$820.00

If an additional extension of time is required, please consider this a petition therefor.

☐ An extension for __ months has already been secured and the fee paid therefor of \$ is deducted from the total fee due for the total months of extension now requested.

☐ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that Applicant has inadvertently overlooked the need for a petition and fee for extension of time.

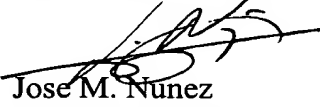
Total Fees Due:

Appeal Brief Fee	\$ <u>510</u> .00
Extension Fee (if any)	\$ _____ .00
Total Fee Due	\$ <u>510</u> .00

☒ Enclosed is Check No. 20439 in the amount of \$510.00

☒ Charge any additional fees or credit any overpayment to Deposit Account No. 50-0850, (Order No. ALTEP072). One copy of this transmittal is enclosed.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP


Jose M. Nunez
Reg. No. 59,979

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
(408) 749-6900
Customer No. 45640

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Nadj et al.

Application No: 09/931,841

Filed: August 16, 2001

For: System and Method for Scheduling
and Arbitrating Events in Computing
and Networking



) Docket No: ALTEP072

) Group Art Unit: 2163

) Examiner: Filipczyk, Marchin R.

) Date: July 11, 2008

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA on July 11, 2008:

Signed: _____

Cynthia C. Lorente

**TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION – 37 CFR 41.37)**

Commissioner for Patents

Box: Board of Patent Appeals & Interferences

Alexandria, VA 22313-1450

Sir:

This Appeal Brief is in furtherance of the Notice of Appeal filed in this case on April 25, 2008.

This application is on behalf of:

☐ Small Entity ☒ Large Entity

Pursuant to 37 CFR 41.20(b)(2) the fee for filing the Appeal Brief is:

☐ \$255.00 (Small Entity) ☒ \$510.00 (Large Entity)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply:

☐ Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a) for the total number of months checked below:

<u>Months</u>	<u>Large Entity</u>	<u>Small Entity</u>
<input type="checkbox"/> one	\$120.00	\$60.00
<input type="checkbox"/> two	\$460.00	\$230.00
<input type="checkbox"/> three	\$1,050.00	\$525.00
<input type="checkbox"/> four	\$1,640.00	\$820.00

If an additional extension of time is required, please consider this a petition therefor.

☐ An extension for __ months has already been secured and the fee paid therefor of \$ is deducted from the total fee due for the total months of extension now requested.

☐ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that Applicant has inadvertently overlooked the need for a petition and fee for extension of time.


Total Fees Due:

Appeal Brief Fee	\$ <u>510</u> .00
Extension Fee (if any)	\$ _____ .00
Total Fee Due	\$ <u>510</u> .00

☒ Enclosed is Check No. 20439 in the amount of \$510.00

☒ Charge any additional fees or credit any overpayment to Deposit Account No. 50-0850, (Order No. ALTEP072). One copy of this transmittal is enclosed.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP


Jose M. Nunez
Reg. No. 59,979

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
(408) 749-6900
Customer No. 45640



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

EX PARTE Paul Nadj et al.

Application for Patent

Filed August 16, 2001

Application No. 09/931,841

FOR:

SYSTEM AND METHOD FOR SCHEDULING AND
ARBITRATING EVENTS IN COMPUTING AND
NETWORKING

APPEAL BRIEF

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450 on July 11, 2008

Signed: _____

Jose M. Nunez

MARTINE PENILLA & GENCARELLA, LLP
Attorneys for Appellants

07/15/2008 CCHAU1 00000017 09931841

01 FC:1402

510.00 OP



TABLE OF CONTENTS

	<u>Page No</u>
I. REAL PARTY IN INTEREST	1
II. RELATED APPEALS AND INTERFERENCES	1
III. STATUS OF CLAIMS	1
IV. STATUS OF AMENDMENTS.....	1
V. SUMMARY OF CLAIMED SUBJECT MATTER	1
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	4
VII. ARGUMENT.....	4
A. Rejection of claims 22-27 under 35 U.S.C. § 112, first paragraph.....	4
1. Claims 22-27	4
B. Rejection of claims 22-27 under 35 U.S.C. § 101	5
1. Claims 22-27	5
C. Rejection of claims 5-10 and 22-27 under 35 U.S.C. § 102(e) over Cochran et al. (U.S. Patent No. 6,701,324).....	6
1. Claims 5-9 and 22-27	6
2. Claim 10	14
D. Conclusion.....	15
VIII. CLAIMS APPENDIX	16
IX. EVIDENCE APPENDIX.....	20
X. RELATED PROCEEDINGS APPENDIX.....	21



I. REAL PARTY IN INTEREST

The real party in interest is Altera Corporation, the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The Appellants are not aware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 5-10, and 22-27 are pending in the subject application. Claims 1-4, and 11-21 have been cancelled. Claims 5-10, and 22-27 have been rejected and are on appeal.

IV. STATUS OF AMENDMENTS

Appellants submitted an amendment on October 1, 2007, in response to a non-Final Office Action mailed on May 29, 2007. This amendment was the last entered amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The subject invention is directed towards a method for high-speed scheduling and arbitration of events for computing and networking is disclosed. The method includes the software and hardware implementation of a unique data structure, known as a pile, for scheduling and arbitration of events. According to the method, events are stored in loosely sorted order in piles, with the next event to be processed residing in the root node of the pile. The pipelining of the insertion and removal of events from the piles allows for simultaneous event removal and next event calculation. The method's inherent parallelisms

thus allow for the automatic rescheduling of removed events for re-execution at a future time, also known as event swapping. The method executes in $O(1)$ time.

Claim 5 defines a method for scheduling events in a computer processing system (*see Abstract; page 31, lines 17-19*), comprising:

identifying queues (*page 31, lines 19-20; Fig. 14*), each of the queues associated with a corresponding priority (*page 32, lines 1-2*), each of the queues including events (*page 31, line 22 to page 32, line 1*);

defining a data structure with a root level having a node group (*See Fig. 6-8; page 21, lines 6-7*), the node group having k number of nodes (*Fig. 8*), each of the k number of nodes sharing a pointer (*Fig. 8; page 21, line 4*), each of the k number of nodes stored contiguously in memory (*Fig 8; page 19, lines 2-3; page 20, lines 6-8*), wherein the k number is equal to a number of multiple queues (*Fig. 8, k is 4, 4 nodes in each node group*);

associating the queues with respective nodes of the data structure (*page 31, lines 17-20*);

assigning a value representing the corresponding priority to the respective nodes (*page 31, line 21 to page 32, line 5*);

determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes (*page 21, lines 9-22*);

selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource (*page 32, line 21 to page 33, line 3; Fig. 16; page 21, lines 9-22*); and

processing the selected one of the events at the processing resource prior to remaining events (*page 32, line 21 to page 33, line 3*).

Claim 22 defines a computer readable medium having program instructions for scheduling events in a computer processing system (*page 31, lines 12-16*), comprising:

program instructions for identifying queues (*page 31, lines 19-20; Fig. 14*), each of the queues associated with a corresponding priority (*page 32, lines 1-2*), each of the queues including events (*page 31, line 22 to page 32, line 1*);

program instructions for defining a data structure with a root level having a node group (*See Fig. 6-8; page 21, lines 6-7*), the node group having k number of nodes (*Fig. 8*), each of the k number of nodes sharing a pointer (*Fig. 8; page 21, line 4*), each of the k number of nodes stored contiguously in memory (*Fig 8; page 19, lines 2-3; page 20, lines 6-8*), wherein the k number is equal to a number of multiple queues (*Fig. 8, k is 4, 4 nodes in each node group*);

program instructions for associating the queues with respective nodes of the data structure (*page 31, lines 17-20*);

program instructions for assigning a value representing the corresponding priority to the respective nodes (*page 31, line 21 to page 32, line 5*);

program instructions for determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes (*page 21, lines 9-22*);

program instructions for selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource; and

program instructions for processing the selected one of the events at the processing resource prior to remaining events (*page 32, line 21 to page 33, line 3*).

It should be appreciated that the above description represents only a summary of the present invention. A more in-depth discussion of the present invention is provided in the Detailed Description section of the application.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds of rejection are presented for review:

- A. Whether claims 22-27 are patentable under 35 U.S.C. § 112, first paragraph
- B. Whether claims 22-27 are patentable under 35 U.S.C. § 101
- C. Whether claims 5-10 and 22-27 are patentable under 35 U.S.C. § 102(e) over Cochran et al. (U.S. Patent No. 6,701,324)

VII. ARGUMENT

Appellants present the following arguments with respect to the rejected claims:

A. Rejection of claims 22-27 under 35 U.S.C. § 112, first paragraph

1. Claims 22-27

i. The feature including “program instructions for processing the selected one of the events at the processing resource prior to remaining events” is described in the Specification as filed.

Claim 22 has been rejected under 35 USC 112, first paragraph, as failing to comply with the written description requirement. Appellants respectfully disagree. The feature including program instructions for processing the selected one of the events at the processing resource prior to remaining events is completely described in the Specification as filed. See for example the following excerpts describing the aforementioned features:

- Fig. 21; “events D, C, and A are dispatched in an order determined by the Scheduling and Arbitration block” (see text in Fig. 21).
- “The use of piles for scheduling and arbitration can be implemented in software using a general purpose processor or in hardware” (page 31, lines 15-16 - emphasis added).
- “Arbitration of the ‘winning’ event, where ‘winning’ is taken to mean the properly chosen next event to process” (page 33, lines 11-12).
- “Appellants contemplate that functional implementations of invention described herein may be implemented equivalently in hardware, software, firmware, and/or other available functional components or building blocks” (page 43, lines 9-11 - emphasis added), etc.

One skilled in the art would appreciate the inventors had full possession of the claimed invention at the time the application was filed.

B. Rejection of claims 22-27 under 35 U.S.C. § 101

1. Claims 22-27

i. Claims 22-27 are directed to statutory subject matter

Claims 22-27 were rejected under 101 as the claims were directed to a non-statutory subject matter. Appellants respectfully disagree. First, claim 22 involves a transformation to a different state. For example, claim 22 defines program instructions for selecting one of the events and processing the selected event, and program instructions for selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource, which creates a new state where the selected event has been transmitted to a processing resource. Therefore, the processing resource in the new state includes the received event, the event residing in the processing resource

memory. This involves a physical transformation of the memory of the processing resource after receiving the new event. Secondly, Appellants define a computer readable medium. See for example: “Appellants contemplate that functional implementations of invention described herein may be implemented equivalently in hardware, software, [and] firmware” (page 43, lines 9-10 - emphasis added). Firmware includes software instructions stored in hardware, thus firmware encompasses a computer readable medium, and the Office’s rejection is improper.

C. Rejection of claims 5-10 and 22-27 under 35 U.S.C. § 102(e) over Cochran et al. (U.S. Patent No. 6,701,324)

1. Claims 5-9 and 22-27

i. Cochran does not teach defining a data structure with a root level having a node group, in the context described in the independent claims

Claim 5 specifies a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory, wherein the k number is equal to a number of multiple queues. The Examiner has asserted that the aforementioned features is anticipated by Cochran in Fig. 1A, items 110, 108n, 106, and 104, col. 5, lines 18-22, and col. 8, lines 41-46. With respect to Fig. 1A, Cochran teaches that “FIGS. 1A and 1B [show] diagrams of a distributed data collection mechanism ... [and that] Distributed data collection mechanism 102 is implemented within a network of data processing systems including endpoints (“E”) 104 coupled via gateways (“G”) 106 to collectors 108a-108n” (Col. 3, lines 29-36 - emphasis added). A network diagram of a data collection mechanism is not a data structure, because a data structure holds data, and more specifically, the claimed data structure includes a root level, node groups, and nodes sharing a pointer.

The other excerpts used by the Examiner teach the following:

"Collector nodes can be connected together to form a collection network topology, and can provide additional functionality such as depoting (caching), bandwidth control, and transfer scheduling. Deployment of collector nodes is controlled by the customer" (col. 5, lines 18-22 - emphasis added).

"Collector 108n also includes a scheduler 424, an active agent which manages the queues 402 and 404 and depot 302. Scheduler 424 services CTOCs 406 from input queue 402, stores and retrieves collection data from depot 302, and propagates collection data upstream to the next collector node." (col. 8, lines 41-46 - emphasis added).

The Examiner has pointed at Routing Manager 110, Collectors 108a-n, Gateways 106, and endpoints 106 to anticipate a data structure, but there is no data structure in Cochran with those elements, as those elements merely describe the hardware of a network topology. Further, the Examiner has pointed at the schedulers in the collector nodes as having queues, thus the Examiner implicitly assumes that the data structures reside in the Collector nodes. With respect to the collectors, Cochran also teaches the following:

"Each collector 108n includes a priority-based queuing mechanism which initially includes two queues: an input queue 402 to store requests for collection from downstream nodes (endpoints or lower level collectors) and an output queue 404 to hold collection requests which were spooled to upstream collectors for pickup. Queues 402 and 404 are maintained in sorted order with the primary sort key being the CTOC priority, which ranges from priority level 0 up to priority level 4." (col. 7, lines 11-19 - emphasis added)

Each collector manages input and output queues for the data being routed through them, as seen in Fig. 4 of Cochran. However, there is no single data structure that would cover all the collectors, and their nodes. Furthermore, selecting items from the queues is done at the collector level, but not considering the data structure as a whole. Claim 5 of the present application also defines selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource. However, there is no mechanism in Cochran to select a highest priority event in the data structure, because there isn't a global data structure.

Appellants understand that the Examiner is giving the broadest interpretation possible to the claims, but pointing to a network topology with distributed data structures aimed at selecting packets independently from the queues in the different nodes, does not anticipate a data structure within a node used to select the highest priority event for transmission. There is not a data structure in Cochran that has all the features of the data structure claimed by the Appellants. In fact, there is not even a data structure in Cochran organized as a tree, only tree-shaped network topologies. Thus, Cochran does not anticipate a data structure with a root level, node groups, and nodes sharing a pointer, as claimed by Appellants.

ii. Cochran does not teach defining a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory

Claim 5 specifies a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory, wherein the k number is equal to a number of multiple queues. The Office has asserted that Cochran teaches the aforementioned in Fig. 1A, item 110, 108n, 106 and 104, col. 5, lines 18-22 and col. 8, lines 41-46. More specifically with respect to the feature of storing nodes contiguously in memory, in the Response to Arguments section the Office has asserted that “[Cochran] refers to it as persistent depot, illustrated in fig. 1A and col. 3, lines 46-67.” Appellants respectfully disagree.

Cochran teaches that “collection data in the form of data packs 408 and data segments 422 are stored on disk within depot 302. Depot 302 maintains an indexed archive of data packs 408 on disk, indexed utilizing the CTOC identifier 410 for the collection data. Depot 302 also implements thread-safety and crash-recovery mechanisms” (col. 8, lines 35-40 - emphasis added). According to Merriam-Webster Online Dictionary, contiguous means “*touching or connected throughout in an unbroken sequence*” (emphasis added). Thus, a depot storing data in data packs and data segments that are indexed, as in Cochran, means that that the data is distributed throughout the disk, therefore, it can not anticipate nodes that are stored contiguously, as claimed by Appellants. The Examiner has provided no articulated reasoning, beyond conclusory statements, as to how a persistent depot discloses contiguous nodes.

It should be noted, that “a group of nodes are co-located in memory such that the nodes may be read with a single long or wide read” (Specification, page 19, lines 2-3). Generally storing data on disk, as in Cochran, does not anticipate nodes stored contiguously in memory in the context claimed by Appellants.

Still yet, it seems that the Office has pointed to Routing Manager 110 from Cochran to anticipate a root level. In the Response to Arguments the Office has asserted that the feature “nodes sharing a pointer” is taught by Cochran because “Gateways and collectors (106 and 108) are interpreted as pointers shared by nodes 104” (see page 5, emphasis added). However, Routing Manager 110 does not have a node group of nodes sharing a pointer, because nodes 104 of Cochran are subordinates of gateway 106 and are not subordinates of Routing Manager 110, as seen in Figure 1A. Furthermore, Gateways and Collectors are not at a root level as illustrated in Figure 1A of Cochran. Therefore, Cochran does not anticipate a root level having a node group, the node group having k

number of nodes, each of the k number of nodes sharing a pointer, as claimed by Appellants. The Examiner's rationale under this line of reasoning is simply illogical.

Furthermore, the Office has not provided an explanation on how Cochran teaches the claimed feature where the k number is equal to a number of multiple queues. Cochran teaches two queues at the Collector, input queue 402 and output queue 404 (see Fig 4). On the other hand, Cochran teaches multiple nodes in the data structure, for example, in Fig. 1A it can be seen that collectors have 3 children, and Gateways have 4 children. Nowhere does Cochran teach that the number of queues is equal to the number of nodes sharing a pointer, and the Office has completely ignored this feature in the rejection and provided no articulated reasoning with rational underpinnings as to where this feature is disclosed. Accordingly, the Office's rejection is improper.

iii. Cochran does not teach assigning a value representing the corresponding priority to the respective nodes

Claim 5 defines assigning a value representing the corresponding priority to the respective nodes, which, according to the Examiner, is anticipated by Cochran in col. 5, line 22, "*scheduling*." Appellants respectfully disagree. Col. 5, line 22 from Cochran merely states that "... control, and transfer scheduling. Deployment of collector ..." (emphasis added). A vague reference to scheduling does not teach assigning a value representing the corresponding priority to the respective nodes, as claimed by Appellants.

The Examiner has asserted that the claimed nodes are taught by Cochran as nodes 104. See the Response to Arguments where the Office has asserted that the feature "nodes sharing a pointer" is "interpreted as pointers shared by nodes 104" (see page 5 of Office Action dated December 28, 2007, emphasis added). The Examiner has also asserted that

Appellants' claimed priority is taught by Cochran as priority 412 inside CTOC 406, where CTOC 406 can be in input queue 402 or output queue 404. For details, see Examiner's rejection of the claimed feature identifying queues, each of the queues associated with a corresponding priority, each of the queues including events. According to the Examiner this feature is taught in items 402, 404, and 406 in Fig. 4 of Cochran (see page 4 of Office Action dated September 28, 2007). There is no explicit mention of where the claimed priority is taught, and Appellants assume that, according to the Examiner, priority is taught as priority 412 of Cochran.

Cochran teaches that "Collection Table of Contents (CTOC) data structures for collection data are received by the collector from the endpoints or downstream collectors and are placed in the input queue, then sorted by the priority within the CTOC" (col. 2, lines 45-49 - emphasis added). The priority in Cochran is used to prioritize CTOCs inside the input and output queues. There is no teaching in Cochran that a priority is assigned to a node, only that priorities are used in the queues inside a node. The Examiner has suggested that a mere mention of scheduling in Cochran teaches assigning a value representing the corresponding priority to the nodes, which is totally unsupported in the teachings of Cochran. The Examiner has failed to show how each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference, therefore the Examiner's rejection is improper.

iv. Cochran does not teach determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes

Further, Claim 5 defines determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes. The Office has asserted that Cochran teaches this feature in fig. 3, 108n, 110, 302 and col. 7, lines 10-19 and lines 53-59. Appellants respectfully disagree. The Office has not provided consistent reasoning on how Cochran anticipates Appellants claims, and more specifically, how Cochran anticipates the features of the nodes claimed by Appellants.

Firstly, in the Response to Arguments the Office has asserted that the feature “nodes sharing a pointer” is “interpreted as pointers shared by nodes 104” (see page 5, emphasis added). Secondly, the Office has asserted that “Cochran teaches storing nodes in contiguous memory and refers to it as a persistent depot” (see page 6). Cochran also teaches that “the primary objective for collectors 108a-108n is to collect data from all corresponding endpoints 104 assigned to route data to the respective collector 108a-108n, and store the received data in a persistent depot until another collector or the ultimate recipient is ready to receive the collected data (col. 3, lines 55-60 - emphasis added). Since the endpoints 104 are the only nodes stored in the persistent depot, the Office has implied again that Appellant’s nodes are anticipated by Cochran through endpoints 104. Additionally, only endpoints 104 from Cochran anticipate nodes as claimed, because other elements in Cochran, such as collectors, are not stored together. However, with respect to the feature of determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes, the Office points to collectors 108, instead to the previously described nodes, and the Office’s rejection is

improper because the collectors do not share the features that are stored together or that they share a pointer. The Office has not provided a coherent argument on how Cochran teaches Appellants' claims, as it refers to different elements in Cochran (endpoints, collectors) to anticipate the same claimed feature (nodes), but not one element in Cochran teaches all the features associated with Appellants' claimed nodes. The Examiner cannot state that Cochran teaches a certain feature and then conveniently ignore that statement when using that feature in an incompatible manner.

v. Cochran does not teach selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource

Claim 5 defines selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource. The Examiner has asserted that Cochran teaches the aforementioned feature in Fig. 4, items 424 and 426, and col. 8, lines 56-65. Appellants respectfully disagree.

As previously discussed, Cochran teaches that the priority is used to sort the CTOCs within the input and output queues 402 and 404. Cochran also teaches that "scheduler 424 [is] an active agent which manages the queues 402 and 404." Therefore, scheduler 424 selects which CTOC is selected first, and not which node having a highest priority is selected next. Scheduler 424 is inside one node and has no visibility of queues in other nodes, thus scheduler 424 can not select a node having a highest priority, and the Examiner's rejection is therefore improper.

2. Claim 10

i. Cochran does not teach resolving conflicts between respective nodes assigned a same value by rotating a pointer among the respective nodes assigned the same value

Dependent claim 10 defines resolving conflicts between respective nodes assigned a same value by rotating a pointer among the respective nodes assigned the same value.

The Office has asserted in the Response to Arguments that Cochran teaches this limitation in the following excerpt:

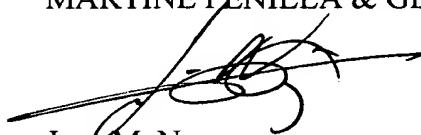
"Referring to FIG. 4, a diagram of a collector in accordance with a preferred embodiment of the present invention is illustrated. The collector is a fundamental element of the distributed data collection service of the present invention, and is responsible for storing and forwarding collected data towards the eventual destination. The collector is a mid-level management object having one instance per host, and providing priority-based queuing of collection requests, depoting of collection data, a crash recovery mechanism for collection data transfers, and multi-threaded transfer scheduling of collection requests in the queues) (col. 6, lines 47-58).

Additionally, the Office has asserted that "based on availability the available node points to the gateway 106 or collector 108 to forward data for collection." This statement is wholly unsupported in the aforementioned excerpt. A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, the identical invention must be shown in as complete detail as contained in the claim. *Richardson v. Suzuki Motor Co.* 868 F.2d 1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. Cochran does not teach "the available node," and more specifically Cochran does not teach "the available node points to the gateway 106 or collector 108." The Offices' assertion is not supported by the teachings of Cochran, and the Office's rejection is thus improper.

D. Conclusion

Independent claim 22 is believed to be patentable for at least the same reasons as with respect to claim 5. The dependent claims are submitted to be patentable for at least the same reasons the independent claims are believed to be patentable. In view of the foregoing reasons, the Appellants submit that each of the claims 5-10 and 22-27 are patentable. Therefore, the Appellants respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's rejections of the claims on appeal.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP

A handwritten signature in black ink, appearing to be 'Jose M. Nunez', written over a horizontal line.

Jose M. Nunez
Reg. No. 59,979

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
Telephone: (408) 749-6900
Facsimile: (408) 749-6901
Customer Number 45640

VIII. CLAIMS APPENDIX

1-4. (Cancelled)

5. A method for scheduling events in a computer processing system, comprising:
- identifying queues, each of the queues associated with a corresponding priority, each of the queues including events;
 - defining a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory, wherein the k number is equal to a number of multiple queues;
 - associating the queues with respective nodes of the data structure;
 - assigning a value representing the corresponding priority to the respective nodes;
 - determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes;
 - selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource; and
 - processing the selected one of the events at the processing resource prior to remaining events.
6. The method of claim 5, further comprising:
- rescheduling the node having the highest priority after selection.

7. The method of claim 6, wherein the method operation of rescheduling the node having the highest priority after selection includes,

determining if the node having the highest priority will be empty after selection.

8. The method of claim 7, further comprising:

if the node having the highest priority will be empty after selection, then the method includes,

removing the value representing the corresponding priority from the node having the highest priority.

9. The method of claim 7, further comprising:

if the node having the highest priority will not be empty after selection, then the method includes,

retaining the value representing the corresponding priority from the node having the highest priority, thereby enabling rescheduling of the node having the highest priority after selection.

10. The method of Claim 5, further comprising:

resolving conflicts between respective nodes assigned a same value by rotating a pointer among the respective nodes assigned the same value.

11-21. (Cancelled)

22. A computer readable medium having program instructions for scheduling events in a computer processing system, comprising:

program instructions for identifying queues, each of the queues associated with a corresponding priority, each of the queues including events;

program instructions for defining a data structure with a root level having a node group, the node group having k number of nodes, each of the k number of nodes sharing a pointer, each of the k number of nodes stored contiguously in memory, wherein the k number is equal to a number of multiple queues;

program instructions for associating the queues with respective nodes of the data structure;

program instructions for assigning a value representing the corresponding priority to the respective nodes;

program instructions for determining a priority between the respective nodes based on respective values representing the corresponding priority to the respective nodes;

program instructions for selecting one of the events corresponding to a node having a highest priority for transmission to a processing resource; and

program instructions for processing the selected one of the events at the processing resource prior to remaining events.

23. The computer readable medium of claim 22, further comprising:
rescheduling the node having the highest priority after selection.

24. The computer readable medium of claim 23, wherein the program instructions for rescheduling the node having the highest priority after selection includes,
determining if the node having the highest priority will be empty after selection.

25. The computer readable medium of claim 24, further comprising:

if the node having the highest priority will be empty after selection, then the computer readable medium includes,

program instructions for removing the value representing the corresponding priority from the node having the highest priority.

26. The computer readable medium of claim 24, further comprising:

if the node having the highest priority will not be empty after selection, then the computer readable medium includes,

program instructions for retaining the value representing the corresponding priority from the node having the highest priority, thereby enabling rescheduling of the node having the highest priority after selection.

27. The computer readable medium of claim 22, further comprising:

program instructions for resolving conflicts between respective nodes assigned a same value by rotating an additional pointer among the respective nodes assigned the same value.

IX. EVIDENCE APPENDIX

There is currently no evidence entered and relied upon in this Appeal.

X. RELATED PROCEEDINGS APPENDIX

There are currently no decisions rendered by a court or the Board in any proceeding identified in the Related Appeals and Interferences section.